Operational Policy Expression and Analysis in the RiverWare Modeling Tool

Timothy Magee*, Edith A Zagona* and Donald Frevert***

*Center for Advanced Decision Support for Water and Environmental Systems, Department of Civil, Architectural, and Environmental Engineering, University of Colorado at Boulder, Campus Box 421, Boulder, CO 80309-0421; PH (303)492-3972; FAX (303) 492-1347; email: magee@colorado.edu, zagona@colorado.edu

** U.S. Bureau of Reclamation, Technical Service Center, D-8510, PO Box 25007, Denver Federal Center, Lakewood, CO 80225-0007; PH (303)445-2473; FAX (303)445-6351; email: dfrevert@do.usbr.gov

Abstract

Operational management of multiobjective river and reservoir systems is facilitated by RiverWare's operational policy modeling features. Rulebased simulation is solved by executing user-specified prioritized rules that are easily created, examined and modified in a syntax-directed graphical editor. The rulebased simulation run analysis utility shows exactly how each policy influenced releases, flows and elevations during the simulation. An optimization solution using pre-emptive linear goal programming is driven by a set of prioritized goals input by the user. Each goal is an objective or a set of constraints on the system; the goals are satisfied in priority order. The Optimization Analysis Tool shows how each goal influenced the optimal solution. Some applications of these tools to real-world problems are described by other papers in this session.

Introduction

Multiobjective river and reservoir system modeling involves using a set of independent policy objectives to drive a solution of releases and/or other control actions that satisfy the objectives as well as possible. Typical objectives reflect such policies as meeting downstream demands, adhering to accepted flood control operations, maintaining depths for navigation, providing flows for water quality and riparian habitat, providing flows and lake levels for recreation, and producing economical hydropower. Conflicting objectives must be given preferences either by weighting factors or prioritization. The solution is generally either a simulation or an optimization algorithm in which the decision variables are reservoir releases and water deliveries. The RiverWare modeling tool provides both simulation and optimization solutions based on prioritized policy objectives.

Both solution approaches solve a network of objects that represent features of the river system such as reservoirs, river reaches, confluences, diversions, and water users. The objects are linked together such that the outflow of an upstream object is linked to the inflow of a downstream object. The same model (river basin network) can be used for both solution approaches. Rulebased Simulation (RBS) is a simulation in which input values are supplied by a prioritized set of rules (Zagona et al., 2001). Rules typically have an IF-THEN-ELSE structure that examines the state of the system in the antecedent, then sets values based on the state of the system. The state of the system could be a prediction or forecast of future states. Higher priority rules may overwrite values set by lower priority rules, but not vice-versa. Linear Goal Programming (LGP) solves the system of physical constraints and prioritized policy constraints and objectives by successively solving linear programming problems defined by the successive policy goals provided by the user (Eschenbach et al., 2001). The goals are either sets of constraints or objectives of the form Minimize(x) or Maximize(x), where x is an expression in terms of the decision variables. If the priority is a set of constraints the constraints are considered "soft" (they may not be attainable) and the objective is to maximize the satisfaction of the constraints. The solution of the problem at each goal cannot further violate the higher priority goals.

RiverWare's two multiobjective modeling frameworks are characterized by the explicit expression of uniquely prioritized policy objectives as input data sets, and by additional output information that shows exactly how the solution was affected by the various policies. However, the two modeling approaches are quite distinct mathematically, both in the form of the policy objectives and in the solution algorithm. This paper examines the similarities and differences between RiverWare's two approaches to multiobjective modeling, and the relative strengths and weaknesses of the two approaches.

Model of the River System Decision Variables

The RiverWare model of the river system consists of objects such as reservoirs, river reaches, diversions, confluences and water users which are linked together to form a network. Each object "contains" physical process models and data to support those models. The links between the objects represent flow continuity, and in some cases, water surface elevation dependencies between the objects.

Simultion. In simulation, the physical process models range from simple linear mass balance equations to complex nonlinear processes such as dynamic routing, flow-varying losses, hydropower generation, release and spill functions, consumptive use, and others. The user tailors the process models to meet the needs of the model and to be consistent with the size of the computational timestep. The tailoring is accomplished by selecting methods from a menu of possible model methods. For example, on a river reach object, a method of routing is selected. The simulation solution is driven by input data that provides enough known values to solve for the unknowns. There are a wide variety of possible combinations of known and unknown values. For example, a reservoir with hydropower releases and spill is characterized by the following process models for mass balance, pool elevation, turbine release and spill:

Storage = Storage (-1) + Inflow – Outflow
Outflow = Turbine Release + Spill (defaults to maximizing turbine flow before spilling)
Pool Elevation = f (Storage)
Energy = f (Turbine Release, Operating Head)
Operating Head = f (Pool Elevation, Tailwater Elevation)
Tailwater Elevation = f (Outflow, and possibly a downstream reservoir's Pool Elevation)

A simulation solution can be arrived at with any of the following combinations of inputs, assuming Storage (t-1) is known:

Inflow and Outflow;
Inflow and Storage;
Inflow and Pool Elevation;
Inflow, Turbine Release, Spill;
Inflow, Outflow, and Spill;
Inflow, Outflow, and Turbine Release;

Storage, Outflow;
Storage, Turbine Release, and Spill;
Storage, Outflow, and Spill;
Storage, Outflow, and Turbine Release;
Pool Elevation, and Outflow;
Pool Elevation, Turbine Release, and Spill;
Pool Elevation, Outflow, and Turbine Release; and
Pool Elevation, Outflow, and Spill.

Thus there are many possible combinations of inputs and outputs. Whenever a reservoir object has enough information to solve, it executes the form of the mass balance equation consistent with the known and unknown variables. If the user specifies too much information, the run terminates in an over-determination error.

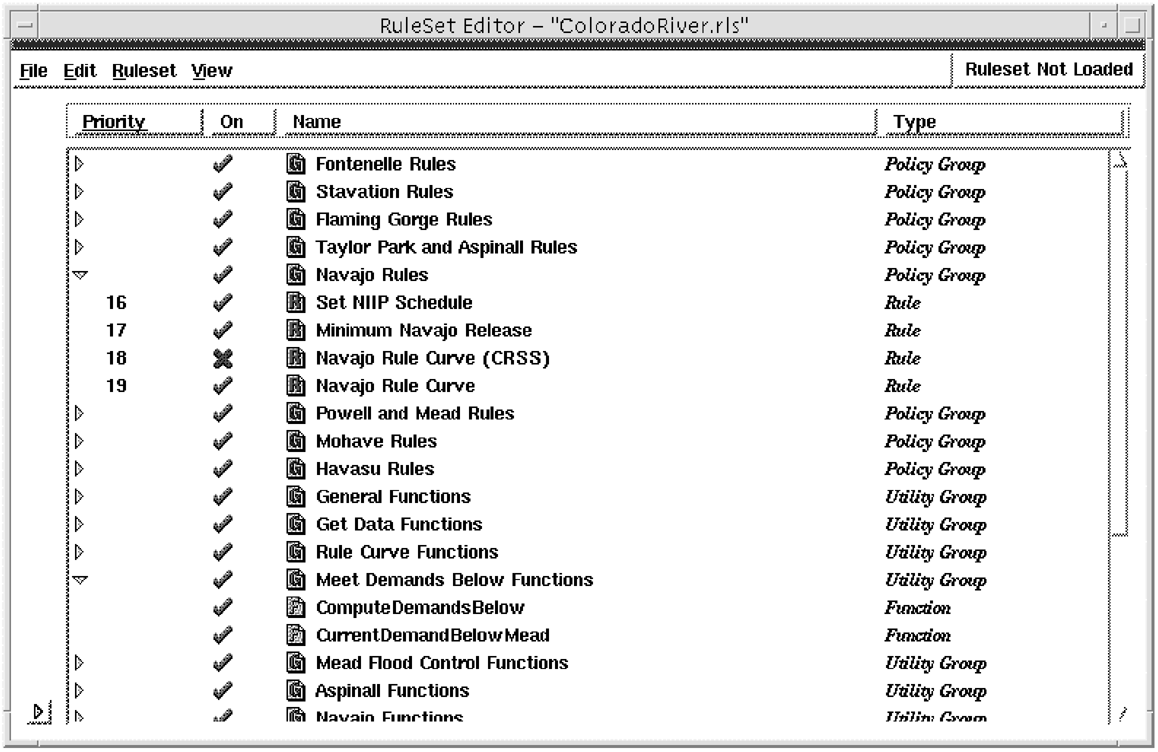
Rulebased Simulation. Rulebased Simulation (RBS) solves in a similar fashion to pure simulation except that before the run, the system is under-determined, i.e., there are not enough known variables to solve the system. Whenever no objects can solve, control is passed to the rule processor, which executes the prioritized rules one at a time. Each rule has the opportunity to examine the state of the system and set values of decision variables according to the logic of the operating policies. As a result, the objects have additional information, which can be used to further the simulation. Even after the objects have solved one or more times, higher priority rules can reset the values of variables that have already been set by lower priority rules or by simulating the effects of other rules.

Linear Goal Programming. In LGP the river system model is simplified. The main decision variables are Storage, Turbine Release, Spill and Outflow from reservoirs and other objects. Other variables such as Inflow, Pool Elevation, Power, Energy, Operating Head, Energy in Storage, and economic variables are defined as linear or piece-wise linear combinations of the decision variables. (The objects generate these linearizations automatically, based on linearization preferences and parameters given by the user.) The first solution is found for the physical constraints including the mass balance and continuity equations for the system. Subsequent solutions are found for each goal beginning with the highest priority. The soft constraints or objectives that make up the goal are translated to objectives that have the form of linear combinations of the decision variables. Each goal is solved without further violating higher priority goals. A post-optimization simulation, driven by the optimal outflows, is used to remove approximations introduced by the optimization. A more detailed explanation is presented in Eschenbach et al., 2001.

Expression of Policy Objectives

For both RBS and LGP the prioritized polices are provided by the user as data to the RiverWare model. The form of the policies is different for the two solution algorithms. For each, the policies are created through a syntax-directed editor. The policies are parsed and interpreted at run time, providing information to direct the solution. For both, individual policies belong to policy groups.

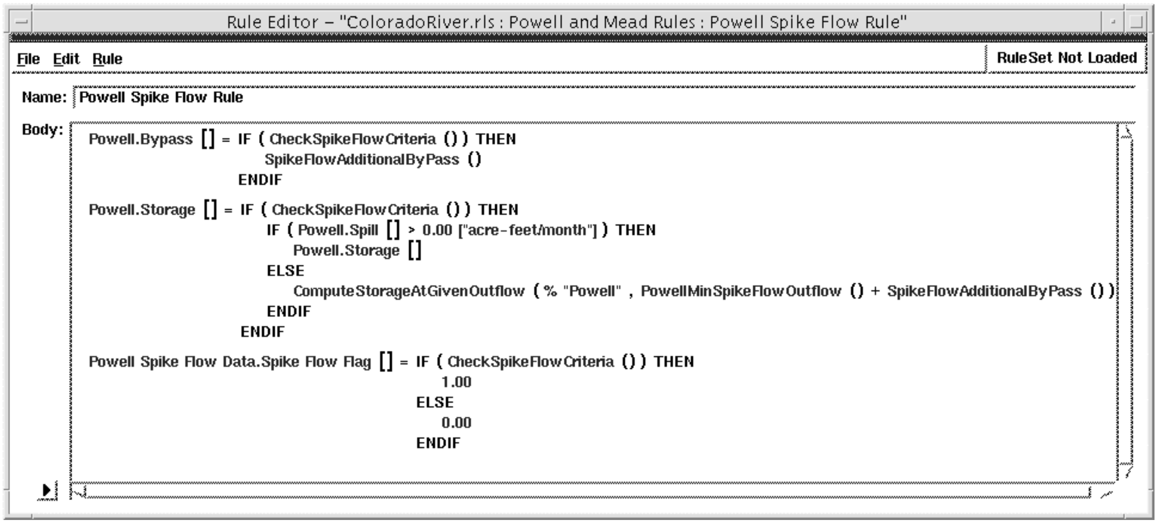
RBS Policy Sets. In RBS, the rules are uniquely prioritized and are grouped by policy set. Figure 1 shows a rule set representing a version of the Colorado River operating policy. The rules are grouped by the part of the system they address. The policy group “Navajo Rules” is expanded to show the rules in the group, rules 16 through 19 are included. Rule 18 displays an ‘X’ in the ‘ON’ column indicating the rule is turned off. Below the rules are functions, defined by the user, to perform certain calculations that are repeated multiple times in the ruleset. The Utility Group named “Meet Demands Below Functions” is expanded to show that it contains two functions.



Priority	On	Name	Type
▷	✓	Fontenelle Rules	*Policy Group*
▷	✓	Stavation Rules	*Policy Group*
▷	✓	Flaming Gorge Rules	*Policy Group*
▷	✓	Taylor Park and Aspinall Rules	*Policy Group*
▽	✓	Navajo Rules	*Policy Group*
16	✓	Set NIIP Schedule	*Rule*
17	✓	Minimum Navajo Release	*Rule*
18	✗	Navajo Rule Curve (CRSS)	*Rule*
19	✓	Navajo Rule Curve	*Rule*
▷	✓	Powell and Mead Rules	*Policy Group*
▷	✓	Mohave Rules	*Policy Group*
▷	✓	Havasu Rules	*Policy Group*
▷	✓	General Functions	*Utility Group*
▷	✓	Get Data Functions	*Utility Group*
▷	✓	Rule Curve Functions	*Utility Group*
▽	✓	Meet Demands Below Functions	*Utility Group*
	✓	ComputeDemandsBelow	*Function*
	✓	CurrentDemandBelowMead	*Function*
▷	✓	Mead Flood Control Functions	*Utility Group*
▷	✓	Aspinall Functions	*Utility Group*
▷	✓	Navajo Functions	*Utility Group*

Figure 1. Rulebased Simulation Policy Set

Each rule consists of assignment statements. Each statement has a RiverWare variable on the left-hand side, and a set of logical statements on the right hand side that evaluates to a number. At the highest level, each rule sets one or more variables. The logic is formulated in the RiverWare Policy Language through the Rule Editor. Figure 2 shows the Rule Editor with a single rule called “Powell Spike Flow Rule” that belongs to the policy group “Powell and Mead Rules.” The rule sets three variables in the model. The logic for the first and second assignments uses a user-defined function, *SpikeFlowAdditionalByPass()*. The second assignment uses a RiverWare-defined function, *ComputeStorageAtGivenOutflow().* The square brackets after the names of variables indicate the value of the variable at the current timestep.



Name: Powell Spike Flow Rule

Body:

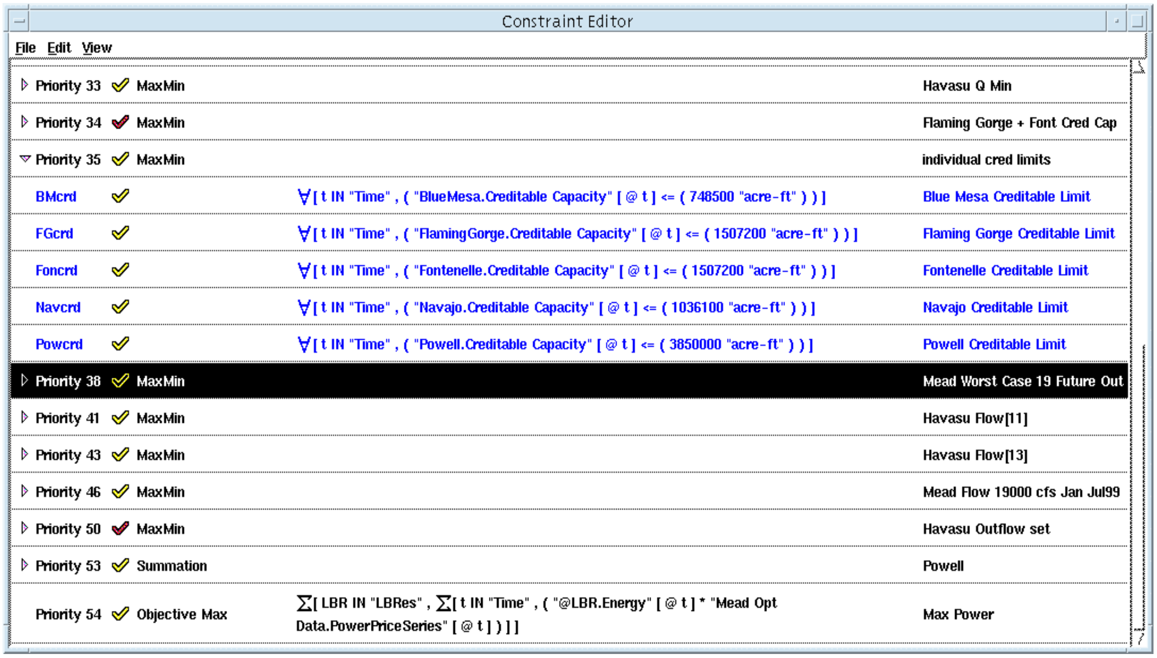
```
Powell.Bypass [] = IF ( CheckSpikeFlowCriteria () ) THEN
                        SpikeFlowAdditionalByPass ()
                    ENDIF

Powell.Storage [] = IF ( CheckSpikeFlowCriteria () ) THEN
                        IF ( Powell.Spill [] > 0.00 ["acre-feet/month"] ) THEN
                            Powell.Storage []
                        ELSE
                            ComputeStorageAtGivenOutflow ( % "Powell" , PowellMinSpikeFlowOutflow () + SpikeFlowAdditionalByPass () )
                        ENDIF
                    ENDIF

Powell Spike Flow Data.Spike Flow Flag [] = IF ( CheckSpikeFlowCriteria () ) THEN
                                                1.00
                                            ELSE
                                                0.00
                                            ENDIF
```

Figure 2. Example of a RiverWare Rule.

LGP Policy Sets. The goal programming solution approach requires the user to specify a set of goals, each of which includes either a system of constraints or a Min or Max type of objective. The goals, which are systems of constraints, may include a number of separate constraints. The software automatically attaches "satisfaction" variables to these constraints to quantify how well the constraints are met, and then formulates an objective that either maximizes the summation of the satisfaction variables, or maximizes the minimum satisfaction variable. The user specifies which of these forms is desired. The maximal satisfaction level is locked in place for lower priority levels, thereby converting the "soft" constraints to "hard" constraints.



Priority 33	✓	MaxMin		Havasu Q Min
Priority 34	✓	MaxMin		Flaming Gorge + Font Cred Cap
Priority 35	✓	MaxMin		individual cred limits
BMcrd	✓		∀[t IN "Time" , ("BlueMesa.Creditable Capacity" [@ t] <= (748500 "acre-ft"))]	Blue Mesa Creditable Limit
FGcrd	✓		∀[t IN "Time" , ("FlamingGorge.Creditable Capacity" [@ t] <= (1507200 "acre-ft"))]	Flaming Gorge Creditable Limit
Foncrd	✓		∀[t IN "Time" , ("Fontenelle.Creditable Capacity" [@ t] <= (1507200 "acre-ft"))]	Fontenelle Creditable Limit
Navcrd	✓		∀[t IN "Time" , ("Navajo.Creditable Capacity" [@ t] <= (1036100 "acre-ft"))]	Navajo Creditable Limit
Powcrd	✓		∀[t IN "Time" , ("Powell.Creditable Capacity" [@ t] <= (3850000 "acre-ft"))]	Powell Creditable Limit
Priority 38	✓	MaxMin		Mead Worst Case 19 Future Out
Priority 41	✓	MaxMin		Havasu Flow[11]
Priority 43	✓	MaxMin		Havasu Flow[13]
Priority 46	✓	MaxMin		Mead Flow 19000 cfs Jan Jul99
Priority 50	✓	MaxMin		Havasu Outflow set
Priority 53	✓	Summation		Powell
Priority 54	✓	Objective Max	∑[LBR IN "LBRes" , ∑[t IN "Time" , ("@LBR.Energy" [@ t] * "Mead Opt Data.PowerPriceSeries" [@ t])]]	Max Power

Figure 3. Linear Goal Programming Policy Set

Figure 3 shows part of a LGP policy set. Several goals are shown. (The ones that are not visible are deactivated.) Goal 38 has been opened to show the constraints. The constraints themselves are formulated in a syntax directed editor.

Managing Policy Objectives over Time and Objects

The two approaches differ fundamentally in how time affects policy. The soft constraints and objectives in LGP may contain decision variables from any time period. In fact, TVA's models frequently have constraints such as guide curves that are applied to all time periods. Simultaneously applying constraints to all time periods allows LGP to make small violations now to prevent large violations later. In contrast, the RBS proceeds strictly forward in time, processing the entire rule set for each timestep before proceeding to the next timestep. This sequential approach defines what a proper rule set should do: assuming all previous values are "known," the rule set, combined with the simulation methods, should set all of the values for the current timestep.

Both approaches have a limited ability to reverse roles relative to their use of time. An optimization model can be restricted to include only policies that apply to the current and previous timesteps and then prioritize the policies by timestep. However, such a policy set eliminates much of the potential to optimize the system. TVA used such an approach as part of the Weekly Scheduling Model (Shane 1988), which motivated some of the functionality in RiverWare. Similarly, rules can include data from future timesteps and be more forward looking than the strict sequencing might appear to be. For example, the Colorado River Simulation System (CRSS) includes "space building" rules which use anticipated inflows and desired future storage capacity to calculate an outflow for the current timestep.

Effect of Policies on Decision Variables

LGP and RBS both restrict the solution space of the decision variables to the physical limitations of the system. LGP does this automatically by applying the physical constraints as the highest priority objectives. For example, the mass balance equations are automatically added at the highest priority. The upper and lower bounds of the variables are also added. In the RBS solution, the rules are responsible for setting values of decision variables that are physically possible. If a rule sets a physically impossible value, the simulation methods catch the error in simulating the effects of the rule and the simulation aborts. The mass balance equations are automatically added after a rule sets a value of a variable. This is accomplished by simulation resulting from the value that is set. One difference is that the simulation methods differentiate between known and unknown variables while a constraint can "solve" for any of the variables included. Simulation can remove this difference by including a method for each possible partitioning of known variables and unknown variables. The enumeration of reservoir mass balance methods presented earlier is a good example of this. In contrast, the simulation could include a reach routing method that will solve for outflows given inflows, without including a method to solve for inflows given outflows.

The analogy between physical constraints and methods also extends to policy. Each rule sets variables based on dependent variables and reduces the solution space in the process. Similarly, after each priority level of a preemptive goal program the solution is constrained to attain the optimal objective value, and this reduces the solution space. In the simplest case the policy immediately reduces the degrees of freedom: a rule sets the value of one or more variables that were previously unknown, or LGP requires one or more variables to be locked into a particular value in order to preserve the optimal objective function value.

Alternatively, a policy may not become active until a lower priority policy is introduced: a lower priority rule may cause a higher priority rule to fire, or a lower priority objective may be limited by the optimal value of a higher priority objective. For example, a high-priority flood control rule checks to see if a flood control situation exists before setting the flood control operation. The rule does not have any effect on the solution unless the flood control “state” exists. Similarly, a high priority elevation constraint in LGP may not be binding until a lower priority constraint reduces outflow.

Some of the details of LGP and RBS are also similar. First, a policy may be redundant: if a rule never fires or a constraint never becomes tight the solution is the same as if the policy never existed. A special case of this is when the solution is completely determined before all of the priorities have been processed; the policies in the remaining priorities are redundant. Second, both the rule set and the constraint set may be underdetermined: at the end of the run RBS may have variables that have never been set, and LGP may have alternate optima. Third, in building a model, the highest priority policies are typically the ones associated with unusual conditions such as extremely high or extremely low reservoir elevations. New users sometimes find it paradoxical that the policies they are most familiar with because they are associated with “normal” operating conditions usually have low priority. For example, one normally would not optimize hydropower until after flood or minimum flow constraints have been satisfied.

Policy Examples Some policies are relatively easy to model using rules or constraints. For example, a minimum flow requirement could be written as a rule:

If Outflow < minimum flow requirement   Then Outflow = minimum flow requirement.

If any lower priority rule sets the outflow to a level less than the requirement then this rule will be triggered and will override the lower priority rule. The corresponding constraint in LGP is

Outflow >= minimum flow requirement, for all time periods.

Lower priority constraints and objectives will be limited by this constraint.

Some policies are easier to model using optimization. For example, balancing reservoir elevations is easier in optimization. An example of balancing is to use the MaxMin metric on the satisfaction of the following constraints with a common priority level:

Reservoir 1. Pool Elevation = Reservoir 1.target
Reservoir 2. Pool Elevation = Reservoir 2.target
Reservoir 3. Pool Elevation = Reservoir 3.target

Writing a similar policy in rules would be considerably more complex.

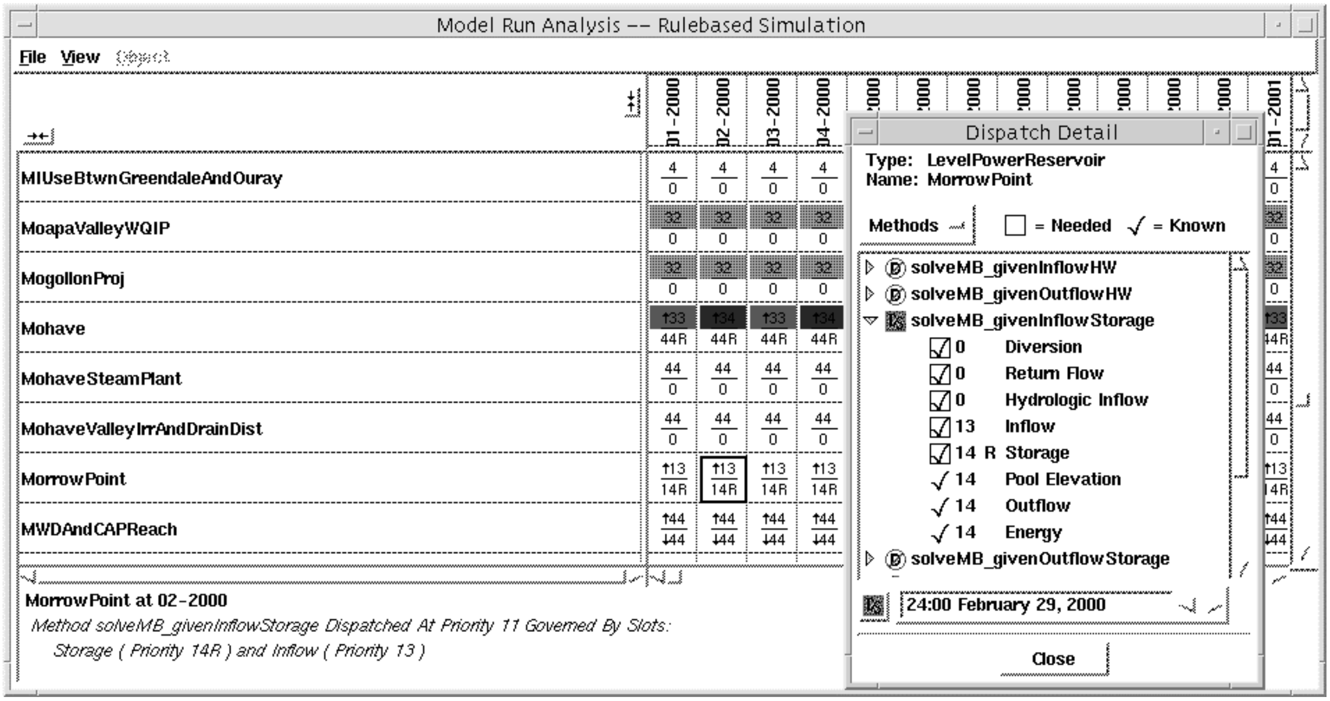
Some policies are relatively easy to write in optimization while impossible to write with rules. Perhaps the most obvious example would be maximizing the economic value of hydropower generation. Details on optimizing hydropower can be found in Zagona and Magee (1999).

Similarly, some rules cannot be replicated within a convex optimization problem. For example, a flood control policy may have an outflow associated with each elevation range of a reservoir. The outflow values for such a policy probably would not be continuous. These discrete jumps in value are beyond convex optimization (but could be modeled in a mixed-integer program). LGP can approximate this policy by allowing outflow to take on a range of values when the reservoir elevation is between two of the elevation policy ranges. Even this kind of policy tends to be more complex in an optimization model than in a rules model (Gilmore et al., 2000).

Analysis of Results

The results of a multiobjective model run include the policies that fixed the final value of each decision variable. RiverWare provides a Run Analysis Utility in which the modeler can view this information for both the RBS and the LGP runs. Each one displays a grid of cells for each object at each timestep.

The RBS cells show priorities of the rules that influenced the solution of the object. For objects such river reaches on which the primary decision variables are inflow and outflow, two values are displayed. The top value is the rule that influenced the value of the inflow (shown by an up arrow) and the bottom value is the rule that influenced the value of the outflow (shown by a down arrow). If the rule directly set the value, an ‘R’ is displayed next to the rule number. Otherwise the value was set as a result of simulation after the indicated rule executed. Reservoirs could additionally be determined by a storage or pool elevation values. The two values that appear on the cell for the reservoir are the values that determined the solution. One is either inflow or outflow (or a variation on outflow such as turbine release + spill), shown by an up or down arrow. A storage or pool elevation value is indicated by no arrow. These features are shown in Figure 4. The user may color-code all priorities in the display of a certain priority, enabling a quick view of the influence of a particular policy over space and time.



	01-2000	02-2000	03-2000	04-2000
MIUseBtwnGreendaleAndOuray	4 0	4 0	4 0	4 0
MoapaValleyWQIP	32 0	32 0	32 0	32 0
MogollonProj	32 0	32 0	32 0	32 0
Mohave	↑33 44R	↑34 44R	↑33 44R	↑34 44R
MohaveSteamPlant	44 0	44 0	44 0	44 0
MohaveValleyIrrAndDrainDist	44 0	44 0	44 0	44 0
MorrowPoint	↑13 14R	↑13 14R	↑13 14R	↑13 14R
MWDAndCAPReach	↑44 ↓44	↑44 ↓44	↑44 ↓44	↑44 ↓44

MorrowPoint at 02-2000

Method solveMB_givenInflowStorage Dispatched At Priority 11 Governed By Slots:
Storage (Priority 14R) and Inflow (Priority 13)

Type: LevelPowerReservoir
Name: MorrowPoint

☐ = Needed ✓ = Known

- solveMB_givenInflowHW
- solveMB_givenOutflowHW
- solveMB_givenInflowStorage
  - ☑ 0 Diversion
  - ☑ 0 Return Flow
  - ☑ 0 Hydrologic Inflow
  - ☑ 13 Inflow
  - ☑ 14 R Storage
  - ✓ 14 Pool Elevation
  - ✓ 14 Outflow
  - ✓ 14 Energy
- solveMB_givenOutflowStorage

24:00 February 29, 2000

Figure 4. Rulebased Simulation Run Analysis Utility

Clicking on one cell opens up a detailed dialog that displays all the decision variables on the object, their priorities a and the combination of inputs and outputs used to solve the object. In Figure 4 the detail dialog shows that Morrow Point Reservoir solved given inflow and storage. Inflow was at priority 13, but not set by a rule. (Hence the value propagated across the link from upstream.). Storage was set by Rule 14 directly. The output variables at the bottom of the list were solved for in the resulting simulation after Rule 14 set the storage, so they also have priority 14. The highlighted cell reflects this information.

While RBS can always definitively state which rule or simulation method is responsible for setting a given decision variable, this is not possible in LGP. Instead, at its most basic level, LGP can only say which variables had to be locked in place to preserve an optimal objective function value, and which additional variables were locked in place as a result of that. Specifically, the additional variables are determined by a subsystem of n equations and n unknowns that exist after the first variables were locked in place. However, when the constraints have special structure more information is available For example, if a policy constraint contains a single decision variable, and that constraint is active, then we can say that this constraint set the variable. Thus, the Optimization Analysis Tool (OAT) has been designed to use this structure when it exists, but otherwise to just list the priority level responsible for setting constraints and variables. While some solutions are intrinsically hard to unravel (when n is large), most solutions in RiverWare can be interpreted using this information.

The top level grid in the OAT is similar to the analysis tool for rules; it has the same grid layout and each grid cell shows the policy priorities responsible for setting the main decision variables for the corresponding object at the corresponding time. For example, a reservoir displays the priorities that set storage and outflow, and a reach displays the priority that set outflow. The same tools that are available in RBS for customizing this display exist in the OAT.

Like RBS, there is a detailed view associated with each cell, which gives additional information about when other decision variables and constraints associated with a cell were set. While the detailed view for rules explains which rule or method set each decision variable, the detailed view for LGP can only do this when special structure exists. Instead, the detailed view presents suggestive information. For example, if turbine release is at turbine capacity this is presented in the detailed view, and a user might reasonable infer that the turbine capacity constraint “set” the turbine release value. However, from a mathematical view one could argue that turbine release was “set” by another constraint in the system of equations and that the turbine release constraint “set” the pool elevation. It is precisely because the first interpretation seems more “natural” that the suggestive information tends to be sufficient to explain the solution.

Summary of Comparison and Application

LGP and RBS in RiverWare solve similar problems using prioritized policy.

1) Both methods incorporate roughly the same decision variables and physical restrictions on those variables, with some simplification and approximation in optimization.
2) Both methods are driven by prioritized policies specified by the user, which gradually reduce the solution space as they become active. In both, policies associated with “unusual” operating conditions have higher priority than “normal” operations.
3) Both methods provide high level and detailed feedback on which policies determined the solution, but the optimization feedback places more responsibility on the user for interpreting solutions.

The methods differ primarily in terms of the kinds of policy that can be specified.

1) RBS requires that policy be applied to earlier time periods before later periods while optimization can solve over time periods simultaneously.
2) Rules differentiate between known variables and unknown variables while constraints and objectives solve for the variables simultaneously.
3) The optimization in RiverWare can only solve over a convex set of solutions, while RBS is not constrained in this sense.

Which method is appropriate for a given application depends primarily on how the desired policies fit with these differences. Given the similarity of the methods some policies can be expressed equally well using either method. For example, either policy can easily specify a minimum in-stream flow requirement. For other policies, either method will work, but one may be a more natural way of expressing the policy. For example, in CRSS the decisions for the current water year are only loosely connected to the next year, and rules are a more natural way to express the policy in this multi-year model. In contrast, TVA's policy includes a preference to balance the storage in several reservoirs and this is more naturally expressed as soft constraints than as rules. Finally, some policies can only be implemented with one approach. For example, optimizing the economic value of hydropower can't be done with rules.

Conclusions

Both RiverWare's LPG and RBS use explicit expression and prioritized policy objectives for a similar set of decision variables, and both approaches provide a similar analysis capability. The differences are primarily in how easily a given set of policies can be expressed (if at all). In principle, one could have a set of policies such that neither approach is suitable for all of the policies. In practice, we have yet to encounter this situation, perhaps because policy makers begin with one approach in mind. Still, the potential for a hybrid approach poses an interesting opportunity for future research.

References

Eschenbach, E., Magee, T., Zagona, E., Goranflo, M. and Shane, R. (2001) "Multiobjective Daily Operations of Reservoir Systems via Goal Programming," to appear in *Journal of Water Resources Planning and Management,* ASCE 127(2).

Gilmore, A., Magee, T., Fulp, T., and Strzepek, K. (2000) "Multiobjective Optimization of the Colorado River," in *2000 Joint Conference on Water Resources Engineering and Water Resources Planning and Management*, ASCE New York, NY.

Shane, R., Waffel, D., Parsly, J., and Goranflo, H. M. (1988) "TVA Weekly Scheduling Model Application Experience," *Computerized Decision Support Systems for Water Managers*, J. Labadie et al (eds), ASCE, New York.

Zagona, E.A., Fulp, T.J., Shane, R., Magee, T. and Goranflo, H.M. (2001) "RiverWare: A Generalized Tool for Complex Reservoir System Modeling," accepted by *Journal of American Water Resources Association.*

Zagona, E. and Magee, T (1999) "Modeling Hydropower in RiverWare " *Waterpower '99,* Proceedings of the International Conference on Hydropower, ASCE New York, NY.